**Amendments To The Claims**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims**:

Claims 1-53 (Canceled)

54. (New) A method of test program generation for a system-under-test, comprising the steps of:

using a primary input stream to generate a sequence of test program instructions for said system-under-test;

defining a set of events, each event in said set having a triggering condition and a predetermined alternate input stream, said primary input stream and said alternate input stream comprising sequences of partially specified program instructions for said system-under-test;

recognizing that said triggering condition of one of said events is satisfied in said primary input stream;

responsively to said step of recognizing, selecting one of said primary input stream and said alternate input stream as a continuation input stream; and

generating subsequent test program instructions using said continuation input stream.

55. (New) The method according to claim 55, wherein said step of defining a set of events is performed by including a portion of said events in an input file that comprises said primary input stream.

56. (New) The method according to claim 54, wherein said step of defining a set of events is performed by storing said set in an event file.

57. (New) The method of claim 54, wherein an occurrence of said triggering condition occurs nonpredeterminedly in said primary input stream.

58. (New) The method of claim 54, wherein said events have priority values, and are processed in order of said priority values.

59. (New) The method according to claim 54, wherein at least a portion of said sequence of test program instructions are randomly generated.

60. (New) A computer software product, including a computer-readable medium in which computer program instructions are stored, which instructions, when read by a computer, cause the computer to perform a method of test program generation for a system-under-test, comprising the steps of:

using a primary input stream to generate a sequence of test program instructions for said system-under-test;

loading a set of events, each event in said set having a triggering condition and a predetermined alternate input stream, said primary input stream and said alternate input stream comprising sequences of partially specified program instructions for said system-under-test;

recognizing that said triggering condition of one of said events is satisfied is satisfied in said primary input stream;

responsively to said step of recognizing, selecting one of said primary input stream and said alternate input stream as a continuation input stream; and

generating subsequent test program instructions using said continuation input stream.

61. (New) The computer software product according to claim 60, wherein said step of defining a set of events is performed by reading a portion of said events from an input file that comprises said primary input stream.

62. (New) The computer software product according to claim 60, wherein said set of events is loaded from an event file.

63. (New) The computer software product of claim 60, wherein an occurrence of said triggering condition occurs nonpredeterminedly in said primary input stream.

64. (New) The computer software product of claim 60, wherein said events have priority values, and are processed in order of said priority values.

65. (New) The computer software product according to claim 60, wherein at least a portion of said sequence of test program instructions are randomly generated.

66. (New) A test program generator, comprising:

a test program generation engine;

a design specification of a target, wherein said design specification comprises a knowledge base, wherein said test program generation engine is coupled to said design specification;

an architectural simulator of said target coupled to said test program generation engine;

said test program generation engine and said architectural simulator being co-operative to perform a method of test program generation for a system-under-test, comprising the steps of:

using a primary input stream to generate a sequence of test program instructions for said system-under-test;

loading a set of events, each event in said set having a triggering condition and a predetermined alternate input stream, said primary input stream and said alternate input stream comprising sequences of partially specified program instructions for said system-under-test;

recognizing that said triggering condition of one of said events is satisfied;

responsively to said step of recognizing, selecting one of said primary input stream and said alternate input stream as a continuation input stream; and

generating subsequent test program instructions using said continuation input stream.

67. (New) The test program generator according to claim 66, wherein a portion of said set of events are included in said primary input stream.

68. (New) The test program generator according to claim 66, wherein said set of events is loaded from an event file.

69. (New) The test program generator according to claim 66, wherein an occurrence of said triggering condition occurs nonpredeterminedly in said primary input stream.

70. (New) The test program generator according to claim 66, wherein said events have priority values, and are processed in order of said priority values.

71. (New) A test program generator, comprising:

a test program generation engine;

a design specification of a target, wherein said design specification comprises a knowledge base, wherein said test program generation engine is coupled to said design specification;

said test program generation engine and being operative to perform a method of test program generation for a system-under-test, comprising the steps of:

using a primary input stream to generate a sequence of test program instructions for said system-under-test;

loading a set of events, each event in said set having a triggering condition and a predetermined alternate input stream, said primary input stream and said alternate input stream comprising sequences of partially specified program instructions for said system-under-test;

recognizing that said triggering condition of one of said events is satisfied;

responsively to said step of recognizing, selecting one of said primary input stream and said alternate input stream as a continuation input stream; and

generating subsequent test program instructions using said continuation input stream.

72. (New) The test program generator according to claim 71, wherein a portion of said set of events are included in said primary input stream.

73. (New) The test program generator according to claim 71, wherein said set of events is loaded from an event file.

74. (New) The test program generator according to claim 71, wherein an occurrence of said triggering condition occurs nonpredeterminedly in said primary input stream.

75. (New) The test program generator according to claim 71, wherein said events have priority values, and are processed in order of said priority values.